MICROPOLIS RES CHANGES TO DRIVE A FRENCH LANGUAGE PRINTER

by A.C.L. Zelmer, Director, International Communications Institute
Box 8268, Station F, EDMONTON, AB T6H 4P1, CANADA

Canada is officially a bilingual country and most federal government documents are available in both french and english. The average western Canadian businessman, however, does not normally use french and the availability of bilingual computer equipment has been minimal until very recently. Unfortunately for me, my work does require the occasional printing of french documents. I could have entered the required accent marks by hand, but this seemed sloppy and looked unprofessional.

The techniques which I used to convert my system to be able to print in both french and english could be used equally well for printing other european language, scientific, or technical characters.

HARDWARE CONSTRAINTS

Over a period of years our equipment has had some changes and modifications. We began with a NorthStar Horizon and Micropolis Mod II disk drives. This is a true Micropolis system, using the Micropolis disk controller and MDOS operating system. The terminal that we presently use is a Cybernex XL-87 with a hardware/software switchable ROM that produces either french or english characters on the screen. Since this is a serial terminal, we do not have a 'video board' on the bus. We continue to use a Digital LA36 dot matrix printer (serial) for draft output and an Epson MX-100 with graphics for better quality output. The MX-100 is still a dot matrix printer, however it has true descenders, 'enhanced' printing styles, and a graphics output mode.

The Cybernex terminal was a replacement for a unilingual Hazeltine 1500 terminal because of our increasing need for the french language character set. The XL-87 'french' character set displays 10 accented characters using single keystrokes. These accented characters replace 10 other characters that are assumed to be unnecessary for work in french. The keyboard is marked with both sets of characters; the keys for the square/wiggily brackets, for instance, have both the bracket markings and the replacement french characters. The characters are thus handled and stored as the corresponding ASCII values for the specified key. The french character representation is a function of the terminal ROM and not the computer itself.

While this terminal can be used with word processing programs to create text, its accented characters typically do not have a corresponding character on the printer used and 'translation' programs must be run to convert the source text before printing. This work is very similar to the special commands that must be inserted in text that is being sent to a typesetting machine, but is cumbersome for quick letters and report production within a normal office.

The Epson printers have the ability to print many foreign language characters by overprinting. Thus a user of WordStar(TM) or a similar word processor could obtain many of the french characters by typing the base letter, an overprint or back-space sequence and the required accent mark. This adds a significant number of keystrokes to a document, encourages errors, and is difficult to proofread on the screen. As well, the printers do not have the ability to produce all of the required accent marks (notably the cedilla and circumflex in french).

MDOS RES

The MDOS RES module contains all permanently resident I/O and Disk Control Routines and associated buffers. For the purpose of this activity, we are interested in the printer routines.

This computer system has two physical printers, the LA36 and the Epson MX-100. In addition, output can be printer to the screen or directed to either the 'french' or 'english' printers. It might be useful to think of the system as having two physical printers, but four LOGICAL printers: the LA36, the CRT screen, the english MX-100 and the french MX-100.

The assembly language program which follows must be executed prior to the use of any printer, physical or logical. It can be built into the RES module itself, or it can be executed just prior to using the printer. In addition, memory location 0003H must be set through a BASIC POKE command, the MDOS ENTR command, or a program action to indicate which logical printer will be used.

Program statements are all shown in upper case letters. Explanations are given in lower case and should not be entered as assembly language program lines. This program works for our specific equipment and needs. It will likely require extensive revision to be useful on your system.

```
*MDOS I/O DRIVER FOR ZELMER  (Programmed by Larry Bauer & Lynn Zelmer)
*          NORTHSTAR HORIZON S-100 BUS WITH Z80
*          CYBERNEX XL-87 SERIAL TERMINAL
*          LA36 SERIAL PRINTER @ 300 BAUD
*          EPSON MX100 PRINTER USING GRAPHICS FOR FRENCH CHARACTERS
                    ;
          LINK 'SYSQ1' ; LINKS TO MDOS ASSEMBLY LANGUAGE EQUATES
          LINK 'SYSQ2' ; PROVIDED BY MICROPOLIS
                    ;
```

An assembly language routine for input/output must know where to find each device, and what values to expect when polling the device. The 'ports' identify where the system should look for specific devices; the 'status masks', etc., indicate what to look for at that port to indicate whether there is a character to be read, or the device is ready to receive a character. The ports, masks, etc., will vary from system to system, but should be available in your computer manual.

```
* THE FOLLOWING PORTS/ETC ARE SPECIFIC TO THIS PARTICULAR SETUP
STAT1:    EQU 03H       ; CONSOLE STATUS PORT
STAT2:    EQU 05H       ; PRINTER / READER STATUS PORT
STAT3:    EQU 06H       ; PARALLEL PRINTER STATUS PORT
MSKO1:    EQU 01H       ; CONSOLE OUTPUT STATUS MASK
MSKO2:    EQU 01H       ; PRINTER / READER OUTPUT STATUS MASK
MSKO3:    EQU 01H       ; PARALLEL PRINTER OUTPUT STATUS MASK
MSKI1:    EQU 02H       ; CONSOLE INPUT STATUS MASK
DATA1:    EQU 02H       ; CONSOLE DATA PORT
DATA2:    EQU 04H       ; PRINTER / READER DATA PORT
DATA3:    EQU 00H       ; PARALLEL PRINTER DATA PORT
                    ;
```

All MDOS application programs extend from 2B00H to the end of contiguous memory. This program starts (or ORGinates) at 2B00H and then jumps to 04E7H for a MDOS 'warmstart'. Note that a number of sections of the code merely overlay current or reserved sections of the supplied RES module. The jump to warmstart is the only actual work that this program accomplishes. The routines and tables are modified for later use by the appropriate application programs.

In other words, executing this program modifies the RES module in memory so that the printer routines will be present for later execution, and returns to the MDOS warmstart for the execution of the next program.

```
          ORG 2B00H
          JMP 04E7H
                    ;
```

The choice of the location 0003H and the terminology IOBYTE conforms to CP/M conventions. The locations from 0000H to 006AH are not used by MDOS, thus they are free to be used as storage areas. This program sets up the printer routines to use the LA36 serial printer with a 0H byte in location 0003H. The MDOS monitor routines can be used to change this memory location at any time with a new value directing the output to one of the other printers.

```
          ORG 0003H     ; LOCATION OF IOBYTE
IOBYTE:   DB 0          ; SEND TO LA36 INITIALLY
                    ;
```

The console I/O jump table is a fixed location in the RES module where MDOS and all applications programs expect to find the addresses of specific console routines. The first 2 bytes will contain the address, for example, of the Console Data INput routine. The use of such a table allows any program to jump to that routine by accessing the known location of the address table.

```
          ORG 04F6H     ; LOCATION OF CONSOLE I/O JMP TABLE
          DW CDIN
          DW CDOUT
          DW CDBRK
          DW CDINIT
                    ;
          ORG 050AH     ; LOCATION OF LIST DEVICE JMP TABLE
          DW LDOUT
          DW LDATN
          DW LDINIT
                    ;
```

MDOS specifies the location of the actual code to implement the printer and console device handling. The addresses can be obtained from the SYSQ files or from a disassembly of your RES module. The following code replaces the originally supplied device handling sections and will likely be identical to your original code for the console routines at least. Printer routines will be new. This code could be

placed anywhere in memory, however it is useful to keep it close to the original RES module location for convenience in making changes.

This particular code had to be modified several times to keep it short enough to fit within the allocated space. Because of the size of the printer routines, they have been moved up to immediately follow the console routines. As supplied there was a small reserve gap between the two sets of routines to allow for future modifications (as here).

```
          ORG 061BH     ; LOCATION OF I/O DRIVER IN MEMORY
                        ;
CDIN:     IN STAT1
          ANI MSKI1
          JZ CDIN
          IN DATA1
          MOV B,A
          RET
                        ;
CDOUT:    IN STAT1
          ANI MSKO1
          JZ CDOUT
          MOV A,B
          OUT DATA1
          RET
                        ;
CDBRK:    IN STAT1
          ANI MSKI1
          XRI MSKI1
          RNZ
          IN DATA1
          MOV B,A
          RET
                        ;
CDINIT:   RET           ; NO INITIALIZATION NEEDED
                        ;
```

The special routines to handle the logical printers begin here. The IOBYTE is checked and the handler jumps to the specific routines specified by the value in IOBYTE. Three of the logical printers essentially follow 'normal' output routines. The routine for the french 'printer' checks every byte of output to see if it matches one of the 'special' characters in its table. If no match occurs, the character is output normally.

Special characters are output according to the table routines. For a number of characters this means a simple <character> <backspace> <overprintcharacter> routine. Characters which cannot be formed by overprinting require shifting into the graphics mode of the MX-100. Graphics mode characters are defined here as being exactly the same width as a normal character, thus we don't have to worry about special line spacing or other non-character problems.

```
LDOUT:    LDA IOBYTE    ; WHERE IS OUTPUT TO GO?
          ANI 0C0H
          JZ LIST2      ; OUTPUT TO SERIAL PRINTER
          CPI 040H      ; OUTPUT TO CRT
          JZ @CDOUT
          CPI 080H      ; ENGLISH TO MX100
          JZ LIST1
          JMP LCONV     ; FRENCH TO MX100
                        ;
LIST1:    IN STAT3      ; GET PARALLEL STATUS
          ANI MSKO3     ; READY FOR OUTPUT?
          JZ LIST1      ; NO, THEN LOOP
          MOV A,B       ; GET CHARACTER TO OUTPUT
          OUT DATA3     ; OUTPUT TO PRINTER
          MVI A,020H
          OUT STAT3     ; RESET STATUS ON MOTHERBOARD
          RET           ; ALL DONE
                        ;
LIST2:    IN STAT2      ; PUT CHARACTER TO PRINTER
          ANI MSKO2     ; LOOP IF NOT READY
          JZ LIST2
          MOV A,B       ; GET CHARACTER INTO A
          OUT DATA2     ; OUTPUT IT
          RET
```

```
                        ;
LCONV:  PUSH H
        PUSH D
        PUSH B
        PUSH PSW
        MVI D,0         ; INITIALIZE D
        LXI H,TABLE     ; INITIALIZE HL TO FRONT OF TABLE
LCONV1: MOV A,M         ; GET FIRST CHARACTER IN TABLE
        CPI 0           ; IS THIS END OF TABLE?
        JNZ LCONV1A     ; NO, THEN CONTINUE
        CALL LIST1      ; YES, THEN PRINT WHATEVER YOU HAVE
        POP PSW
        POP B
        POP D
        POP H
        RET             ; ALL DONE, SO RETURN
LCONV1A: CMP B          ; NO, THEN IS IT CHARACTER TO REPLACE?
        JZ LCONV2       ; YES, THEN REPLACE IT
        INX H           ; NO, THEN GET LENGTH OF ENTRY
        MOV E,M
        DAD D           ; GO TO NEXT TABLE ENTRY
        INX H
        JMP LCONV1
LCONV2: INX H           ; GET LENGTH OF STRING TO BE OUTPUT
        MOV C,M
LCONV3: INX H           ; GET CHARACTER TO BE OUTPUT
        MOV B,M
        CALL LIST1      ; OUTPUT IT
        DCR C           ; DONE YET?
        JNZ LCONV3      ; NO, THEN GET THE NEXT CHARACTER
        POP PSW
        POP B
        POP D
        POP H
        RET             ; YES, THEN ALL DONE
                        ;
```

The first character, lower case 'a' grave, is an overprint character. The table has one byte for the keyboard character (040H, the @ sign), one for the replacement character count (03H), and three bytes to be output (one of which is a backspace).

The second character, lower case 'a' circumflex, is a character over-printed with a graphic. The circumflex is similar to an up-arrow sign on some system, however the Epson up-arrow was unsatisfactory and the bit pattern shown is used as a replacement. As with the first character, the lower case 'a' and the backspace is output almost normally. The printer is then shifted into the graphics mode and a bit pattern printed according to the following bytes output. Each byte corresponds to the pattern for one vertical row of dots and can easily be determined by drawing the required character on graph paper.

Other characters are formed similarly.

```
TABLE:  DB 040H         ; A GRAVE
        DB 03H
        DB 061H,08H,084H
        DB 05BH         ; A CIRCUMFLEX
        DB 18
        DB 061H,08H,1BH,'L',12,00,00,00,00,040H,080H,040H,00,00,00,00,00,00
        DB 05CH         ; C CEDILLA
        DB 18
        DB 063H,08H,1BH,'L',12,00,00,00,00,01,00,01,00,00,00,00,00,00
        DB 5DH          ; E CIRCUMFLEX
        DB 18
        DB 065H,08H,1BH,'L',12,00,00,00,00,00,040H,080H,040H,00,00,00,00,00
        DB 5EH          ; I CIRCUMFLEX
        DB 18
        DB 069H,08H,1BH,'L',12,00,00,00,00,040H,080H,040H,00,00,00,00,00,00
        DB 060H         ; O CIRCUMFLEX
        DB 18
        DB 06FH,08H,1BH,'L',12,00,00,00,00,00,040H,080H,040H,00,00,00,00,00
        DB 07BH         ; E ACUTE
        DB 3
        DB 65H,08H,083H
```

```
        DB 07CH         ; U GRAVE
        DB 3
        DB 075H,08H,84H
        DB 07DH         ; E GRAVE
        DB 3
        DB 065H,08H,084H
        DB 07EH         ; U CIRCUMFLEX
        DB 18
        DB 075H,08H,1BH,'L',12,00,00,00,00,040H,080H,040H,00,00,00,00,00,00
        DB 0            ; ALL DONE SIGNAL
                        ;
LDATN:  XRA A           ; SET Z FLAG
LDINIT: RET
                        ;
        END
```

The procedures indicated have been used to implement a foreign language character set on an Epson MX-100 printer using an I/O routine contained within the RES module of MDOS. It would be quite feasible to define a complete character set in this way (although the larger table would have to be relocated elsewhere in memory) to simulate a proportional spaced printer. Thus users with an occasional need for decent, proportional spaced typing, or for draft quality output of proportional spaced typing could obtain this from the MX-100. Special mathematical characters, superscripts and subscripts, etc. are also possibilities.

.....

VECTOR JOINS WITH DUAL SYSTEMS CONTROL CORPORATION

Reprinted from the Vector News

Thousand Oaks, California, April 4, 1985 -- Vector Graphic today announced that it has terminated further discussions with Mr. Eric Tang with respect to an agreement in principle previously announced by the Company pursuant to which Mr. Tang, representing a group of related companies, would guarantee certain secured indebtedness of Vector and acquire a major stock and voting position in the Company.

Vector Graphic further announced that it has signed today an agreement in principle with Dual Systems Control Corpoiration of Berkeley, California for the merger of Dual into Vector.

Dual Systems Control Corporation is a privately held corporation which manufactures and markets a 16/32 bit Unix based multi-user computer system and board level products including boards incorporating sophisticated IO and SMD controller technology. Incorporated in January 1980, Dual has sold systems domestically and internationally, concentrating on the engineering, scientific and educational markets. Dual, which has been profitable since 1982, had net income after taxes of approximately $1,000,000 on gross revenues of approximately $10,000,000 on an unaudited basis for the calendar year ended December 31, 1984.

Under such agreement in principle, all shares of Common Stock of Dual outstanding at the date of the merger and underlying options and warrants to purchase Dual Common Stock would be converted into and exchanged for shares of Common Stock of Vector and comparable options and warrants for Vector Common Stock. Immediately following the merger, the former shareholders, option and warrantholders of Dual will hold approximately 77% of the total number of the then issued and outstanding shares of Common Stock of Vector after assuming full exercise of such options and warrants. The remaining 23% of the outstanding shares of Common Stock of Vector will be held by the current Vector shareholders and various other individuals and entities expected to receive a significant number of shares of Vector Common Stock in connection with the satisfaction of certain conditions to the merger which require third-party approvals. No assurance can be given that such approvals can be obtained. The merger is further conditioned upon, among other things, the execution of a definitive agreement of merger, approval by the Boards of Directors of Vector and Dual, and approvals of shareholders of both companies.

The agreement in principle further contemplates that for the interim period prior to the consummation of the merger, the day-to-day operations of Vector will be administered by Dual under the supervision of the Vector Board pursuant to the terms of a Management Contract and a Joint Operating Plan.

As part of the management contract, Bernard Horn, President and CEO of Vector, and Richard A. Hahn, Senior Vice President of Marketing and Sales, will resign as officers of Vector and Mr. Hahn will resign as a director of Vector. Mr. Horn will continue to serve as a director and assist in the management transition.

Mr. Horn stated that the contemplated merger would combine Dual's management talent, technical expertise, and product concepts in the Unix multi-user area with Vector's product, dealer and user base to position the combined companies for future growth.

.....

USING MICROPOLIS DISKCOPY UNDER CP/M

by Burks Smith of Datasmith, Inc.
P.O. Box 8036, Shawnee Mission KS 66208 - (913) 381-9118

I have long been aware that the Micropolis DISKCOPY program is about one third faster (approximately 2 minutes versus 3 minutes) than the Vector Graphic BACKUP.COM program, which is distributed on their CP/M master disks. Both programs make binary copies of the source disk, so both will copy disks formatted either for CP/M or MDOS. Since the Micropolis version is faster, I have been routinely using it to copy CP/M disks, but it is sometimes inconvenient to get out the MDOS system disk and boot it up when working with CP/M.

It is possible to put Micropolis DISKCOPY on a CP/M disk with no ill effects. This is because the MDOS operating system loads in low memory, while CP/M loads in high memory. The two programs do not share any memory, so they can co-exist at the same time. Under some circumstances, it is even possible to run MDOS as an application under CP/M.

Micropolis DISKCOPY is an overlay program that overwrites the MDOS console command processor, leaving only the peripheral and disk I/O routines. It uses all available memory and jumps to the coldboot address (in ROM) when it is through. DISKCOPY uses addresses from 2B1H through 1DC0H for itself and the MDOS I/O routines, and these addresses lie within CP/M's applications area.

To get DISKCOPY on a CP/M disk, you need a computer with a ROM monitor that will allow you to manipulate memory and jump to a specific address without benefit of an operating system. All Vector Graphic monitors have this ability, and the instructions below apply to a Vector system. Actually it is possible for DDT or RAID to do the same thing, but it's somewhat more complicated. Proceed as follows:

1. Boot CP/M in the normal way. The operating system is loaded into high memory.

2. Press the hardware RESET button to get to the monitor (don't turn the machine off).

3. Replace your CP/M disk with an MDOS disk and boot MDOS by pressing B at the monitor prompt. When MDOS signs on, type DISKCOPY. At this point, both DISKCOPY and CP/M are in memory, both undamaged and perfectly capable of running.

4. Press the hardware RESET button again to get back to the monitor. All CP/M programs are expected to have their first executable instruction at location 100H, so we have to "patch" the DISKCOPY image so CP/M will correctly execute it. To do this, we put a jump instruction at location 100H that sends execution to the MDOS warmstart address at 4E7H, which is DISKCOPY's entry point. DISKCOPY doesn't use location 100H, so this won't hurt anything. From the monitor, enter the following sequence:

P 0100
then enter
C3 E7 04
then press the ESC key

5. DISKCOPY is now ready to be saved on a CP/M disk. Put your CP/M system disk back in the drive and enter the monitor command:

G000

This will start execution at location zero, which happens to be a jump instruction to CP/M's warmstart address that is still there from when you first booted CP/M. The disk drive will select and CP/M's A> prompt will reappear.

6. Finally enter the CP/M command:

SAVE 29 DISKCOPY.COM

This saves 29 "pages" of 256 bytes each to your disk as a .COM file.

To use diskcopy, just enter its name as a CP/M command. It will sign on as usual and ask for source and destination drives. Remember that since this is an MDOS program, drive "A" is called "0" and drive "B" is called "1". Since diskcopy uses all available memory in the copying process, it will write over CP/M when it runs. You must put a bootable disk in drive A before exiting the program.

.....

LANGUAGE COMPILERS: SOME PRACTICAL OBSERVATIONS -- Part 1: BASIC/Z

by A.C.L. Zelmer, Director, International Communications Institute
Box 8268, Station F, Edmonton, AB T6H 4P1, Canada

I am not a programmer in any real sense of the word. I have, however, learned to do a fair amount of 'ad hoc' programming to develop small utilities that fill in the gaps in commercial programs, or routines that allow me to operate my system(s) more conveniently. I have therefore developed a number of programs using Snobol on a mainframe computer and BASIC on the micros.

More recently, I have flirted with a number of compilers in an attempt to overcome some of the shortcomings of BASIC. In this series of three short articles I will look at the advantages and disadvantages of three language compilers for the CP/M operating system. An early version of the first of these compilers was also available for use under MDOS.

WHAT IS A COMPILER?

We probably need to start by looking at the computer system and how it translates our programs into commands meaningful to the computer.

BASIC is normally an INTERPRETED language. The computer translates each command on a line by line basis as the program is executed. This interpretation at run-time leads to easily changed programs, but lacks execution speed. Interpreted languages are high level languages and can be transferred to other computer systems with minimal changes.

Assembly language code is ASSEMBLED by the computer into an executable form only once. The code can then be reused as often as required without the time-consuming line-by-line translation of an interpreted language. Assembly language programs are typically the fastest type of programs, however they are difficult to prepare because of the low level of the language (mnemonics only) and the necessity to know the characteristics of the computer that they will be used on. Each CPU chip has its own assembly language mnemonics and codes. Because the code is assembled into the program by the developer, it is often very difficult to make even minor changes to the program.

COMPILED languages are also translated into executable form only once during the program preparation. Compiled languages are higher level languages than assembly language mnemonics, and may use exact copies of a high level interpreted language's syntax and commands. An interpreted language can then be used to design and test a program that is later compiled for safer use and distribution. To a user such as myself, the two biggest advantages of a compiled language are the ease of programming because of the high level language involved and the security of use that comes from the user being unable (accidentally or otherwise) to change the program during operation.

Some compilers actually generate assembly language code that can be modified and assembled using the normal assembly language program development tools. Others generate code that requires a run-time library of standard routines (input, output, functions, etc.) to be executed at the same time to operate. These systems typically have a RUN.COM file that must be present on the disk at the same time as the program being executed (this program can be shared by any number of programs present on the disk). Still other compilers produce stand-alone .COM files in what appears to the user to be a single step.

THE PROJECT:

For the purpose of this comparison, I have chosen a very simple utility program that switches my Epson MX-100 from normal to condensed printing mode. The program could be expanded to use any of the MX features, however it has only the one function for clarity. A short message is printed to the screen and the user's response solicited before changing the printer, so the program demonstrates a number of the statements common to many of our programs.

The MX printer shifts from normal to condensed mode upon receipt of a '<CTRL>O' character (decimal 15, octal 017), and returns to normal mode with a '<CTRL>R' character (decimal 18, octal 022). The system where this program is being used has the printer as the standard CP/M LST: device.

BASIC/Z:

Steven Guralnick has described the BASIC/Z compiler in the BASIC/Z CORNER so I won't describe the system at any length. BASIC/Z uses its own editor to check the syntax of program lines as they are being entered and compiles the program to a file that is executed using a run-time module, the RUN/Z library. Programs can be linked to this module to create a single program file for distribution.

Note that BASIC/Z does not have any commands that direct the output to the printer. Instead, all output is sent to the "print stream" using standard PRINT statements. Print re-direction statements may change

the final destination of the print stream, the LPRINTER statement in the program below directs the output to the printer. The UPCASE statement similarly forces all input to upper case.

Also note that BASIC/Z does not have the variable naming restrictions of Micropolis BASIC.

The RUN/Z library requires 24K of disk space. This program compiled to a 2K file. Thus approximately 26K of disk space is required to store the completed program. The program, PRINTER.BZO is executed with the command:

RZ PRINTER

at the system prompt or from within the BASIC/Z editor. `BASIC/Z programs are relatively easy to develop for a person with a knowledge of almost any dialect of BASIC. The syntax checker eliminates simple spelling errors and contributes to programming speed. It took less than one-half an hour to prepare this program, including the testing required.

```
2 ! PRINTER.BZS -                       Change the MX-100 print mode
4 !                                     A.C.L. ZELMER, Feb 85
6 !                                     for Basic/z compiler
10 PRINT "PRINTER"
20 PRINT "This program shifts the MX-100 printer.  ";
30 PRINT "The printer must be ON;"
40 PRINT
50 PRINT "    for normal type           enter N"
60 PRINT "    for condensed type        enter C"
70 PRINT "    to eXit (quit)            enter X"
75 UPCASE
80 INPUT "and press the <ENTER> or <RETURN> key.  :"; RESPONSE$
100 RESPONSE$ = LEFT$(RESPONSE$,1)
110 IF RESPONSE$ = "N" THEN LPRINTER: PRINT CHR$(18): STOP
120 IF RESPONSE$ = "C" THEN LPRINTER: PRINT CHR$(15): STOP
130 IF RESPONSE$ = "X" THEN STOP
140 PRINT "Please enter one of the specified responses."
150 PRINT "Press the <ENTER> or <RETURN> key to continue."
160 INPUT RESPONSE$
170 GOTO 10
```

Next time I'll discuss the TURBO Pascal compiler from Borland.

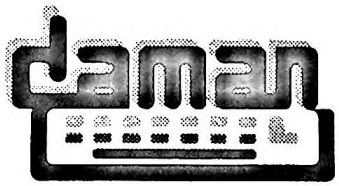
.....

CLASSIFIED
==========

FOR SALE - My hardware was destroyed, but I still have manuals for System-B and approximately 30 single-sided, double-density, 16-sector, disks and an operating system. Will sell all for $75. Also have software such as Wordstar, Multiplan, D-Graph, D-Base.

Roland Chalton, 143 Commercial Ave., Kirkland WA 98033 (206) 641-3693 or (206) 827-9034.

.....



STAR PRINTERS
==========

STAR Micronics' introduction of its new "S" series printers represents advanced printer technology at industry trend-setting prices. Some of the significant features include 20% increased print speed and Near Letter Quality (NLQ) printing as standard on all models. STAR has also combined the standard and IBM-PC versions into one printer. This means that a single printer interfaces with your S-100 system, IBM-PC, Apple, Commodore, and all compatibles. DIP switches are accessible from the printer's exterior for easiest access. The user manuals are very complete. Star has a full one-year warranty on all their products.

For all printing needs, letters, memos, reports - any printing job that fits on a 10" piece of paper - Star has four exciting new printers for you. STAR also has three sophisticated new 15" printers for spreadsheets and other business uses.

SR-10/15

- 200 characters per second (cps) draft quality, 50 cps near-letter quality.
- 16K buffer on 15", 2K on 10" (expandable to 6K)
- Friction and tractor paper feed, single sheet inserter, short form tear-off, replaceable cartridge ribbon, all standard
- Hex Dump (for debugging)
- 100 % compatibility with IBM-PC, Commodore, Apple and all compatibles
- Easy access format switches, ultra high resolution graphics
- Bidirectional logic seeking printing

.....

SD-10/15

- 160 characters per second (cps) draft quality, 40 cps near-letter quality.
- 16K buffer on 15", 2K on 10" (expandable to 6K)
- Friction and tractor paper feed, replaceable cartridge ribbon, all standard
- Hex Dump (for debugging)
- 100 % compatibility with IBM-PC, Commodore, Apple and all compatibles
- Easy access format switches, ultra high resolution graphics
- Bidirectional logic seeking printing

.....

SG-10/15

- 120 characters per second (cps) draft quality, 30 cps near-letter quality.
- 16K buffer on 15", 2K on 10" (expandable to 6K)
- Friction and tractor paper feed standard
- Hex Dump (for debugging)
- 100 % compatibility with IBM-PC, Commodore, Apple and all compatibles
- Easy access format switches, ultra high resolution graphics
- Bidirectional logic seeking printing

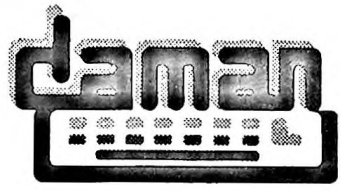
.....

SB-10

- 144 characters per second (cps) draft quality, 60 cps near-letter quality.
- Optional 128K memory
- 24-wire print head
- Friction and tractor paper feed, short form tear-off, replaceable cartridge ribbon, all standard
- 100 % compatibility with IBM-PC, Commodore, Apple and all compatibles
- Adaptable for integrated software
- Easy access format switches, ultra high resolution graphics

.....

DAMAN 604 Springwood Circle, Huntsville, Alabama 35803 (205) 881-1697
Computer Programs • Systems Enhancements • Data Management • Support Services



STAR PRINTERS

Guaranteed satisfaction.
Return within 30 days for full refund.

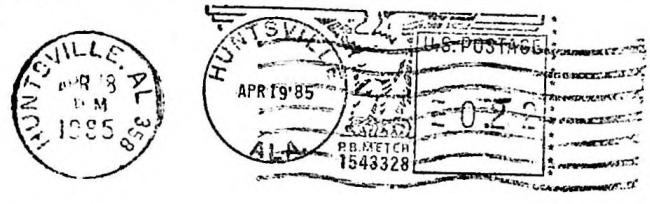
Printer	List Price	DAMAN Price-1	DAMAN Price-2
Stx-80	$199	$195	$201
SG-10	$299	$269	$277
SG-15	$499	$412	$424
SD-10	$449	$376	$387
SD-15	$599	$483	$498
SR-10	$649	$519	$535
SR-15	$799	$626	$645
SB-10	$949	$733	$755
Powertype	$499	$354	$365

Accessories	List Price	DAMAN Price-1	DAMAN Price-2
SG Serial Interface	$ 59	$ 56	$ 58
Serial Interface with 4K Buffer	$119	$111	$114
Serial Printer Interface Cable		$ 25	$ 26
Powertype Tractor	$ 59	$ 56	$ 58

Price-1 = Pre-paid. Price-2 = Charge.

****
* MICROPOLIS/VECTOR GRAPHIC USERS GROUP NEWSLETTER *
* Published Monthly by DAMAN, 604 Springwood Circle, Huntsville AL 35803 *
* Subscription rates: North America; $18/year Elsewhere; $25/year, Airmailed *
*
* For Assistance with general information, MUG/Vector Graphic, DAMAN hardware *
* & software, Micropolis Basic, Basic/z, Micropolis drives & Micropolis parts-*
*
* Call Lynn or Buzz at (205)881-1697 during the Central Times of 9 AM to 9 PM.*
*
****

MICROPOLIS/VECTOR GRAPHIC
USERS GROUP
A Division of DAMAN
604 Springwood Circle
Huntsville AL 35803
(205) 881-1697



FIRST CLASS MAIL

FIRST CLASS MAIL